

# A model for integrating learning object repository resources into web videoconference services

Aldo Gordillo, Enrique Barra, Daniel Gallego and Juan Quemada

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

Avenida Complutense 30, 28040, Madrid, Spain

{agordillo, ebarra, dgallego, jqquemada}@dit.upm.es

**Abstract**— Reusing Learning Objects saves time and reduces development costs. Hence, achieving their interoperability in multiple contexts is essential when creating a Learning Object Repository. On the other hand, novel web videoconference services are available due to technological advancements. Several benefits can be gained by integrating Learning Objects into these services. For instance, they can allow sharing, co-viewing and synchronized co-browsing of these resources at the same time that provide real time communication. However, several efforts need to be undertaken to achieve the interoperability with these systems. In this paper, we propose a model to integrate the resources of the Learning Object Repositories into web videoconference services. The experience of applying this model in a real e-Learning scenario achieving interoperability with a web videoconference service is also described.

**Keywords**— online learning; videoconference; learning objects; interoperability; repository

## I. INTRODUCTION

Digital Learning Objects are the building blocks of e-Learning activities. A Digital Learning Object (hereafter just LO) may be as simple as a paragraph of text or an image, or can be a more complex object like a quiz or an educational game. Due to the immensity of learning resources on the Internet, Learning Object Repositories (LORs) play a key role by facilitating their discovery and reuse. The repurposing of LOs in multiple contexts is essential to enable cost-effective development and foster collaboration between organizations. Hence, achieving their interoperability is one of the main challenges when creating a LOR.

On the other hand, the adoption of novel HTML5 [1] technologies and the growth of cloud computing services are enabling the development of innovative web videoconference services. These new technology solutions provide high quality services at low cost as well as unprecedented easy access since users just need a web browser to establish a real time communication.

These new services can offer huge benefits for distance and blended learning. They allow the sharing, co-viewing and synchronized co-browsing of educational resources at the same time that provide real time communication. However, to take advantage of these benefits, the LOs have to be integrated with the videoconference services.

Basic LOs like a simple image are very easy to integrate, but more complex ones, which can have a high degree of interactivity, need more sophisticated solutions. There are several interoperability standards for LORs, but they are usually more focused on achieving interoperability among them than on increasing the ability of their resources to be reused. On the other hand, some standards have been defined to integrate LOs in external environments such as Learning Management Systems (LMSs), but they were not designed with web videoconference services in mind. Therefore it is not possible to take full advantage of the power of these tools using existing standard solutions.

In this paper we propose a novel model to integrate the LOs of a LOR into web videoconference services. We also present the experience of implementing this model in a real e-Learning scenario, in which we achieve the interoperability with a web videoconference service called MashMeTV (<http://mashme.tv>).

The rest of the paper is organized as follows. The next section reviews related work of LORs focused on interoperability. Section 3 explains the model. Section 4 presents the scenario in which it has been applied as well as the use case. The last section finishes with some concluding remarks together with an outlook on future work.

## II. RELATED WORK

The Learning Object Metadata (LOM) standard [2] defines a Learning Object as “any entity, digital or non digital that may be used for learning, education or training”. However, in e-Learning this term usually covers only digital entities. The main aim of providing educational resources as LOs is to facilitate reuse [3]. Reusing LOs instead of repeatedly authoring them can lead to several benefits such as savings in time, reducing development cost and even enhancing the quality [4]. A key factor in improving reusability and interoperability of LOs is metadata, where their information is specified including their structure, properties and educational characteristics.

Usually, LOs are created using e-Learning authoring tools and stored, searched and accessed using LORs. A LOR stores both LOs and their metadata, either by storing them physically together or by presenting a combined repository to the outside world [5]. LORs also allow users to search and retrieve LOs. To provide the most possible resources, they take LOs from

other LORs using different methods such as federated searches or harvesting. Metadata plays an important role in facilitating the retrieval process since LORs typically support advanced searches based on LOM properties (e.g. subject, target age). The most widespread standard to define the metadata scheme is LOM, although some LORs have implemented their own scheme. LORs also provide several services to users beyond searching and storing such as favorites, recommendations of LOs based on user's history and private repositories.

Some different interpretations exist for the term interoperability. It is often considered as the capability of one system to communicate and interact with others. But in this context, interoperability can be also defined as the ability of "enabling information that originates in one context to be used in another in ways that are as highly automated as possible" [6]. In this paper we always refer to interoperability according to this second definition. Then, a crucial factor of a LOR is to provide interoperable LOs. Or in other words, to provide LOs that can be integrated automatically with LMSs and third party websites. Several approaches exist to achieve interoperability with different systems. First off, most e-Learning authoring tools build LOs conforming to some e-Learning standard such as SCORM (Sharable Content Object Reference Model) [7] or AICC (Aviation Industry Computer-Based Training Committee) allowing their reuse in LMSs. Furthermore, many LORs offer the possibility of exporting their resources to some of these standards.

Lastly, we must take into account that not all integrations are equally strong. Some approaches (e.g. SCORM) allow communication between the LOs and the context in which they are running while others do not. In the context of videoconference services, regarding the achieved degree of interoperability, the LOs can be *shared* (i.e. each participant sees his/her own isolated instance of the LO), *co-viewed* (i.e. all participants see the same LO) or *co-browsed* (i.e. all participants can also realize actions over the co-viewed LO synchronously). For instance, the Bridgit videoconferencing tool [8] achieves LO interoperability by using computer desktop sharing. Therefore the LO can be co-viewed since all participants can see the same LO at the same time, but co-browsing is not possible since the only participant who can interact with the LO is the one who is sharing the computer desktop. Some other videoconference services offer computer desktop sharing (e.g. Google Hangouts) but all of them have this limitation.

### III. MODEL

This model has been designed with Web Videoconference Services (hereafter WVSSs) in mind. Its main aim is to satisfy the main use cases of a scenario where a WVS is used for distance and blended learning: sharing, co-viewing and synchronized co-browsing of LOs.

The designed solution is based on the idea that if a LOR can export its resources to e-Learning standards to integrate them in LMSs, then it also would be possible to export these resources using certain conventions to allow their integration into WVSSs.

According to this model, all LOs are delivered as web resources including a lightweight component that provides an API (Application Programming Interface), which enables the communication between the context in which LOs will be used and themselves. Fig. 1 illustrates this idea that will be explained in detail along the next sections.

#### A. Delivering interoperable Learning Objects

The LOR provides a main function: Request/Deliver LO. This function is in charge of attending the requests of LOs and delivering them enabling their reuse in the way that the interoperability can be achieved as highly automated as possible. Several steps are carried out to attend a LO request:

1. If the LO is not in a web-ready format it should be adapted. This adaption can be done when the LO is stored in the LOR or on the fly in a less efficient way. Hence, if an image is requested, the LOR will deliver a simple HTML page with the image embedded in it.
2. Besides the LO, a lightweight JavaScript API (hereafter LO API) will be attached to the HTML page in order to detect the events triggered as a response of the user's actions. This API also provides functions to replicate these actions. This step makes no sense for non-interactive resources (e.g. an image) and then in those cases can be skipped. However, if an interactive LO is requested like an HTML5 video, the API allows to detect when a user plays or stops the video as well as to replicate any of these actions.
3. Finally, the HTML page will be delivered to the application or website that requested the LO. This application will integrate the LO using an HTML element called *iframe*.

#### B. Introducing the *iframe* element

Before continuing with the explanation of the model, we present in this section a brief summary of the *iframe* element. An *iframe* is an HTML element that is used to insert or integrate other HTML documents or resources inside a website. For instance, a YouTube video can be embedded in a website using an *iframe*. The *iframe* element was standardized by the W3C (World Wide Web Consortium) and is supported in all modern browsers. Other kind of HTML elements can be used to embed web content in a website (e.g. *embed* or *object*), but the *iframe* is the only one that allows bidirectional communication between the website and the embedded object in a standardized way. Following with the previous example, using the YouTube API it is possible to detect when the user clicks on the play button or to play the video automatically without the user intervention.

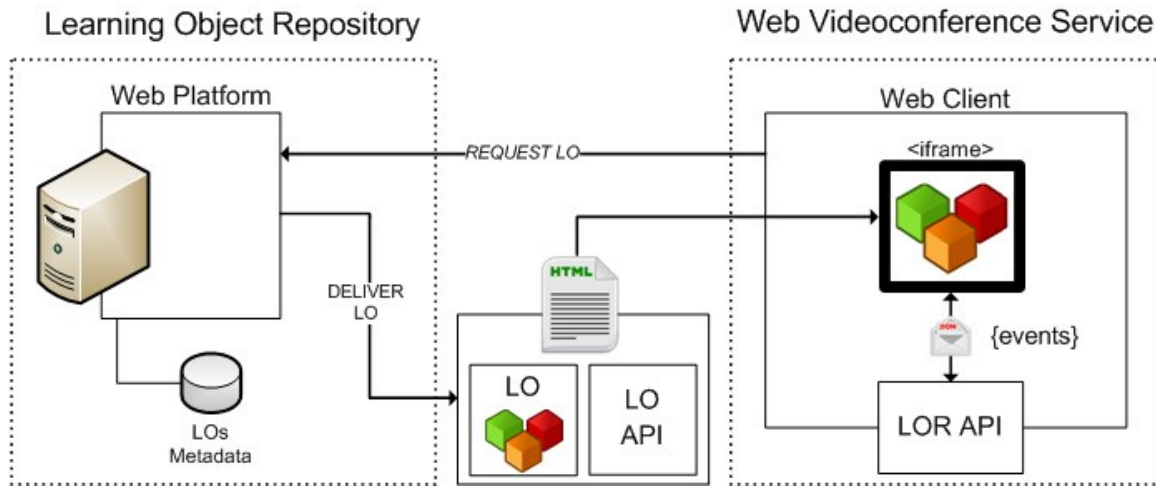


Fig. 1. LOR model to achieve interoperability with web videoconference services

### C. Delivering particular LOs

In section 3A we saw that using a common API it is possible to automatically detect and replicate the events of standardized web elements such as an HTML5 video. However, this approach is not valid for particular LOs since they have their own specific actions. For example, the actions that users can perform in a slideshow (e.g. advance slides) are quite different that the actions they can do in an educational game (e.g. jump or shoot). Then, each kind of particular LO needs to provide its own LO API. As an example, Slideshare (<http://www.slideshare.net>) provides an API to interact with its presentations that specifies six actions: *jumpTo*, *next*, *previous*, *first*, *last* and *getCurrentSlide*.

Taken into account that these resources are usually created through authoring tools, the model specifies a convention that can be followed by these tools in order to create LOs that can be automatically integrated into WVSs. This way, teachers can create their own LOs (or adapt their existing ones) and share and co-browse them in a videoconference session.

Our solution requires the authoring tool to define each action that can be performed in the LO together with its corresponding triggered event. It needs to be able to replicate the different actions and detect their corresponding events. Therefore, in this case authoring tools will be in charge of providing the LO API. Furthermore, it is advisable that they include the possible events in the LO metadata or at least use the “Interactivity Level” parameter specified by LOM which describes the degree of interactivity of the LO.

For example, if a slideshow authoring tool wants to integrate their presentations into WVSs following this model, it may provide a LO API considering one action *jumpTo* to go to a specific slide, and its corresponding event *onSlideEnter* triggered when the user enters in a new slide. This way, if a user goes to slide 5, the *onSlideEnter* event will be triggered with the parameter 5. On the other hand, if the *jumpTo* method is invoked with the parameter 5 the slideshow will advance to

slide 5. The key idea relies on the fact that in both cases, the final state of the slideshow is the same.

### D. Delivering Learning Object events

In the previous sections we have explained how to detect the user actions for standard web resources and particular LOs as well as how to replicate these actions based on the events triggered. In this section, we are going to explain how these actions are delivered in a WVS environment. First off, the WVS should include the API of the LOR (hereafter LOR API) the same way as it includes third party APIs from external services (i.e. videos and slideshows). Fig. 1 also illustrates this fact. This API includes two functions: *onMessage* to receive the messages from the LO API and *sendMessage* to send them. These methods enable communication between the LOR API and the iframe in which the LO is inserted. This way, a bidirectional communication channel is built between the LO and the web videoconference client (hereafter just client) through the LOR API.

Fig. 2 represents through an example the different actions that are carried out to deliver the LO events. In this example, Alice and Bob are participating in a videoconference session talking and co-browsing a slideshow presentation. They are commenting the sixth slide. In a certain moment, Alice clicks on a button to go to the seventh slide.

1. The Alice’s slideshow (i.e. the slideshow that Alice is seeing) advances to slide 7. As a consequence an event is triggered and handled by the LO API. Then, it composes a message including all relevant information about the event, in other words, all the information needed to replicate the event. Finally, the LO API notifies the LOR API.
2. The LOR API receives the message and delivers it to the client, which will send it to the Messaging Server of the WVS.

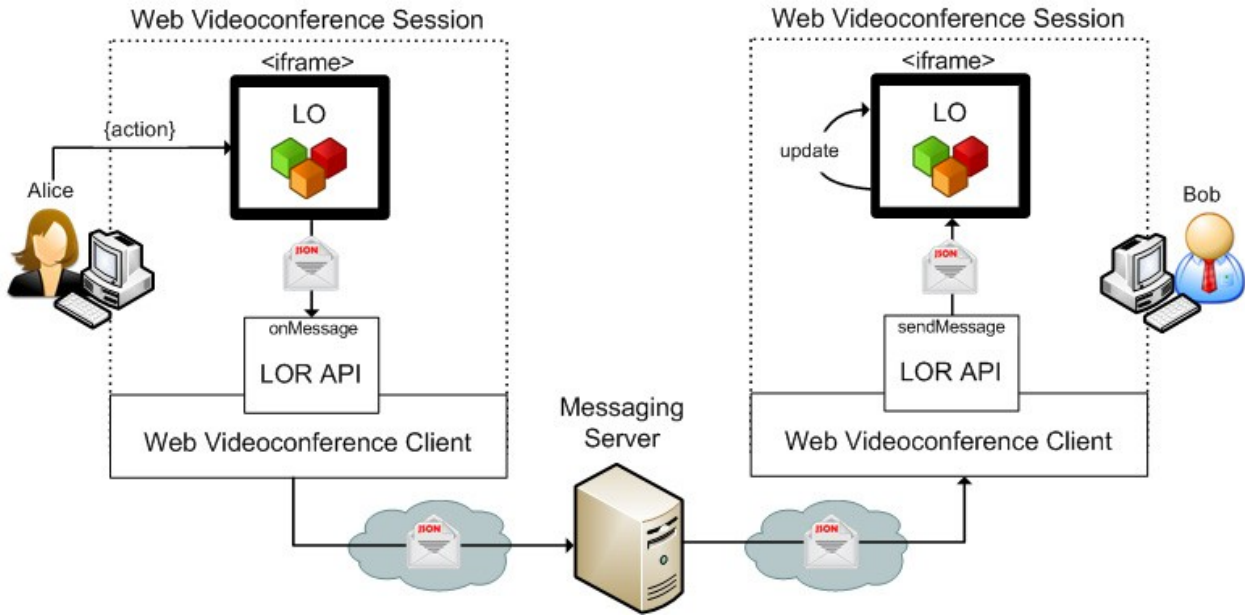


Fig. 2. Delivering Learning Object events

3. The Messaging Server do not have to read or understand the message generated by the LO, it just have to broadcast the message to the rest of the participants (in this case just Bob) through the network using its usual data channel.
4. After the broadcasting, Bob's client receives the message and sends it to the LO API.
5. When Bob's LO receives the message, it composes the original event and replicates the action. Therefore, Bob's slideshow also advances to slide 7. Hence, the final state of Alice's and Bob's slideshows is the same.

This is an example of co-viewing, but considering that Bob can also perform actions in the slideshow in the same way Alice does, co-browsing is also provided.

#### E. Achieving synchronized co-browsing

Previous section shows an example of co-browsing, but it is not synchronous. Considering again the initial situation of the previous example where the slideshow is in the sixth slide. If Alice clicks on the next button and at the same time Bob clicks on the previous button, Alice notifies that she advances to slide 7 while Bob notifies the he advances to slide 5. As a consequence, the final state will be different: Alice will be in the slide 5 since she receives Bob's notification, while Bob will be in the slide 7 due to Alice's notification. This problem is called "state synchronization" and is one of the biggest concerns in the development of distributed real time applications that requires synchronization (such as online multiplayer games). In the field of games, it has been defined as "the problem of maintaining the same game state information on each of the players' instance of the game and generating the actions of each player belonging to the same game instance" [9]. This definition is perfectly valid in our

context, replacing game for LO and players for participants. After all, a LO can be an educational game.

Due to this problem, achieving synchronized co-browsing requires to use some distributed synchronization mechanism to order the events produced by the different LO instances. Several synchronization mechanisms and algorithms can be used to deal with this problem such as the logical clocks mechanism, which can be provided by different algorithms like the Lamport timestamps [10] or the vector clock [11]. Any synchronization mechanism that guarantees the ordering of the events is valid to apply the proposed model. However, some mechanisms do not operate completely in the client and need to implement logic in the Messaging Server, and therefore they may be unviable in common scenarios where access to the Messaging Server is not possible.

Another concern that should be taken into account is latency (i.e. time delay experienced), since it is the most critical network aspect related to the user experience of online distributed real time applications (e.g. multiplayer games) [12]. Fortunately, several latency avoidance mechanisms exist to alleviate the latency problem such as dead reckoning [12] and lockstep synchronization [13] that can be useful in this context.

Finally, to illustrate all these concepts, we give below a straightforward example of a solution to achieve synchronized co-browsing. In this solution a new operation mode is provided by the LO API. In this mode each time a user tries to perform an action, instead of allowing him/her to do the action and then notify it to the rest of the participants, the action will be prevented but the notification will be sent. Then, the WVS has to broadcast the event not only to the rest of the participants, but also to the user who originates the broadcasted event. This way, all participants will receive the event approximately at the same time depending on their latencies.

After that, a simple implementation of a lockstep event-locking mechanism is used to achieve synchronized co-browsing. In this mechanism the server distributes the events periodically. During each period or time slot, the clients can send their events to the server to be broadcasted, but the messages that arrive too late will be considered lost. Hence the state is updated every period. Consequently, the server just has to deal with the “event collisions”: when two incompatible events are received in the same time slot. In this example we use a straightforward approach that consists of allowing just one event per time slot and in case of collision choosing one randomly and discard the rest.

This implementation has some limitations. For example, new participants (i.e. those that enter in the videoconference session when several actions have been performed over the LO) will not be able to compose the full LO state. So, a last improvement can also be made. The server can store all actions performed over the LO, and deliver them to the new participants that join later.

Finally, back again to the example where Alice and Bob click a button at the same time but considering now the solution described above, the process will occur in the following way:

1. After the clicks, Alice’s and Bob’s slideshows remain static but one notification per event is sent.
2. The two messages are delivered to their corresponding clients and sent to the Messaging Server.
3. As the two messages will arrive in the same time slot, the Messaging Server choose one randomly, for example, the Alice’s message, and broadcast it to all participants (Alice and Bob).
4. When the message arrives, each client sends it to its corresponding LO.
5. The Alice’s action is performed in both slideshows. As a consequence, the final state of Alice’s and Bob’s slideshows is the same again.

#### F. The Iframe Gateway Method

In previous sections we have seen how to integrate LOs into WVSs. However, in those examples, a little effort from the WVS was always required to achieve interoperability. It needs to include the LOR API and provides message broadcasting. In some occasions, a WVS may be interested in performing other actions like encrypting messages, but these actions are not mandatory.

In some cases a WVS (e.g. Google Hangouts) may have its own API to provide third party web applications with functionalities like messaging (allowing communication between application instances) or access to users’ information. This model takes advantage of this fact to make possible to achieve interoperability with a WVS without requiring its cooperation when it provides its own API (WVS API). This way it is not necessary for the WVS to include the LOR API and the integration process can be completely done by the LOR.

To add this improvement, an extra step is introduced in the Request/Deliver LO function described in Section 3A. Instead of directly delivering the web page with the embedded LO, the LOR inserts it in an iframe and delivers a new HTML page that includes this iframe and that will act as a gateway between the WVS and the LO. We call this method the “Iframe Gateway”. Its operation is represented in Fig. 3. The LO still communicates with the LOR API. But in this occasion, the LOR API will deliver the messages to the *iframeGateway*, which will communicate with the web videoconference client through the public API of the WVS. This is a good strategy since it keeps intact both WVS and LOR APIs and can be implemented with little effort.

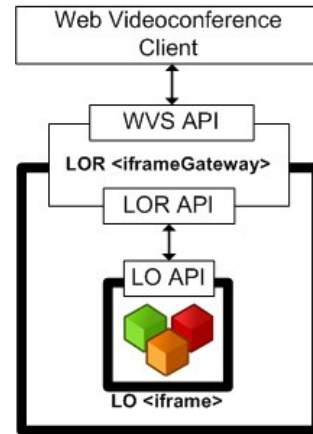


Fig. 3. Iframe Gateway Method

#### IV. ViSH: A CASE STUDY

The GLOBAL excursion (Extended Curriculum for Science Infrastructure Online) project [14] is a European project which main aim is to enrich science teaching in European schools. Via a central web portal, called the Virtual Science Hub (ViSH), GLOBAL excursion provides scientists, teachers and their pupils a package of activities, materials and tools for enabling the integration of e-Infrastructures into school curricula. The ViSH platform includes a social network, an e-Learning authoring tool and a LOR that stores all the LOs created or uploaded by users and institutions.

The authoring tool allows ViSH users to create a novel LO called Virtual Excursion [15], [16]. Virtual Excursions are presented as rich interactive slideshows. They can contain diverse resources: multimedia files, e-Infrastructure resources (e.g. a webcam or a remote pendulum), web games and flashcards among others. Flashcards are resources presented as a background image with several “hot zones” identified by arrows where the user can touch and see additional contents that the teacher has previously tagged [17]. The e-Infrastructure resources are provided by scientific institutions and allow students to virtually visit different infrastructures such as research laboratories or natural parks. However, sometimes the guidance of an expert is required to enjoy these resources. For example, a scientist may be required to control a microscope and explain the concepts to the students.



The selected solution in the GLOBAL excursion project to connect scientist and classrooms in real time was to share the Virtual Excursions in a WVS. This was achieved by applying the presented model both to the ViSH LOR as well as to the authoring tool used to create the Virtual Excursions.

#### A. MashMeTV

In this web platform people can share many resources such as pictures, Slideshare or PDF presentations, blackboards, maps and YouTube videos. All elements are synchronized with the room (i.e. conference session) ensuring that all participants see the same at the same time. For this reason, MashMeTV was selected as the most appropriate WVS to co-browse synchronously the ViSH LOR resources. A new button was added in all excursions of the ViSH platform in order to init a MashMeTV videoconference session sharing the desired resource. This way, all participants can explore synchronously the Virtual Excursions and at the same time communicate in real time among them. As an example, Fig. 4 shows a flashcard shared in a MashMeTV session.

The first step to achieve integration was applying the model to the existing e-Learning authoring tool. As a result, the authoring tool provides a LO API which is capable of detecting all actions performed over a Virtual Excursion and replicating these actions based on their corresponding events. Since MashMeTV provides its own API, the integration with the ViSH LOR was performed successfully using the Iframe Gateway method. In this case the ViSH LOR acted as *iframeGateway* to connect the LO API with the official API of MashMeTV.

Since this integration, MashMeTV has been used regularly to hold meetings with scientists, students and teachers, sharing and co-browsing a lot of Virtual Excursions. Moreover, teachers of the ViSH Platform were fascinated with the possibility of creating their own learning resources and sharing them easily through a videoconference service. They were delighted because this feature allows them to guide their pupils in the exploration of different Virtual Excursions while they have real time expert guidance.

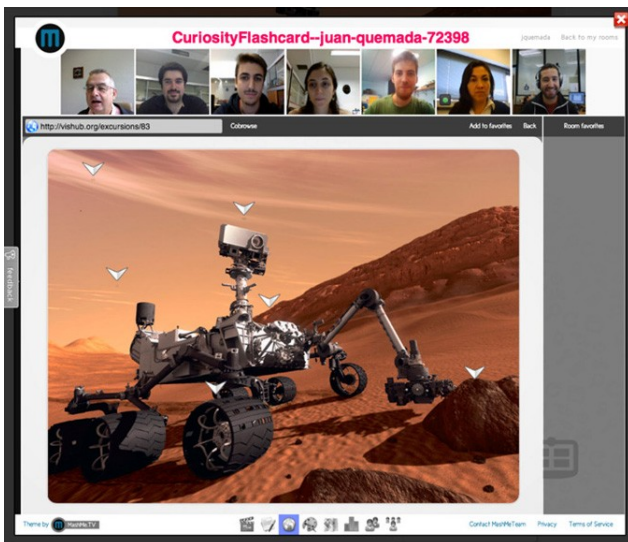


Fig. 4. Flashcard shared through MashMeTV

## V. CONCLUSIONS AND FUTURE WORK

We have presented a model to integrate LOs into WVSs. Achieving interoperability with WVSs can lead to several benefits such as co-viewing and synchronized co-browsing of the LOs. However, some efforts need to be undertaken to make this integration possible: LORs should provide LOs in web-ready format, e-Learning authoring tools should provide a suitable API for their LOs, and WVSs should provide an own API or include the LOR API and take charge of messaging. In case of using the WVS API, the LOR should act as a gateway between the LO and WVS APIs. Synchronized co-browsing also requires implementing some synchronization mechanism. Finally, the experience of applying the model in a real e-Learning scenario achieving interoperability with a WVS has been also presented.

New learning opportunities to increase students' readiness and engagement arise when integrating LOs into videoconference systems as a consequence of the combination of real time communication and synchronized co-browsing. A good example of this is the GLOBAL excursion project, in which the integration of e-Infrastructures resources (e.g. a microscope) with a videoconference service is used to connect scientists and classrooms.

Creating interoperable LOs that can be shared and co-browsed in videoconference services may be very easy for educators if e-Learning authoring tools follow a specific model and automate the process.

The immediate next step of this work consists of validate the model in more WVSs such as Google Hangouts.

In addition, this research opens new lines of investigation as a result of the integration of LOs into videoconference systems. We plan to use this model in the development of educational online multiplayer games that take advantage of videoconference features, where players can communicate and play in real time collaborating or competing among them.

Finally, based on the lessons learned from developing the Iframe Gateway method, we plan to adapt this model to build a LO Gateway with the main aim of achieving the interoperability of standardized e-Learning resources (e.g. SCORM packages) with web videoconference services.

## VI. ACKNOWLEDGMENT

We wish to acknowledge our gratitude and appreciation to all the GLOBAL excursion project partners, and each one of the project team members, for their contribution during the development of various ideas and concepts presented in this paper. We also would like to thank the SAAN project (TIN2010-19138) for funding this work.

## REFERENCES

- [1] W3C, "HTML5 specification." [Online]. Available: <http://www.w3.org/html/wg/drafts/html/master/Overview.html>.
- [2] IEEE LTSC, "Draft Standard for Learning Object Metadata," 2002.
- [3] D. A. Wiley, "Learning Object Design and Sequencing Theory," Brigham Young University, 2000.
- [4] P. Mohan and C. Brooks, "Learning Objects on the Semantic Web," in *Proceedings of the 3rd IEEE International Conference on Advanced Technologies*, 2003, pp. 195–199.

- [5] F. Neven and E. Duval, "Reusable Learning Objects: a Survey of LOM-Based Repositories," vol. 68, no. 4, pp. 291–294, 2002.
- [6] G. Rust and M. Bide, "The indecs metadata framework: Principles, model and data dictionary," 2000.
- [7] "Advanced Distributed Learning (ADL), SCORM 2004 4th Edition," 2004. [Online]. Available: <http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th>.
- [8] "Using learning objects with Bridgit, video conferencing and interactive whiteboards to connect classrooms," 2008.
- [9] A. Spurling, "QoS Issues for Multiplayer Gaming." [Online]. Available: <http://users.cs.cf.ac.uk/O.F.Rana/data-comms/gaming.pdf>.
- [10] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," vol. 21, no. 7, 1978.
- [11] C. J. Fidge, "Timestamps in Message-Passing Systems That Preserve the Partial Ordering," in *Proceedings of the 11th Australian Computer Science Conference (ACSC'88)*, 1988, vol. 10, no. 1, pp. 56–66.
- [12] C. Westermarck, "Mobile Multiplayer Gaming," KTH Information and Communication Technology, 2007.
- [13] N. E. Baughman and B. Neil, "Cheat-Proof Payout for Centralized and Distributed Online Games," in *Proceedings of the IEEE INFOCOM 2001 Conference*, 2001, pp. 1–11.
- [14] T. Holocher-ertl, B. Kieslinger, and C. M. Fabian, "Linking schools with science: How innovative tools can increase the effectiveness of science teaching in the classroom," 2012.
- [15] A. Gordillo, E. Barra, and J. Quemada, "Enhancing K-12 science education through a multi-device web tool to facilitate content integration and e-Infrastructure access," in *Proceedings of the 7th International Technology, Education and Development Conference (INTED 2013)*, 2013.
- [16] B. Kieslinger, T. Holocher, C. M. Fabian, D. Gallego, S. Aguirre, E. Barra, and G. Mihai, "Virtual Excursions: a New Way to Explore Science in Class," in *Proceedings of the 2th International Conference on New Perspectives in Science Education (NPSE 2013)*, 2013.
- [17] E. Barra, D. Gallego, S. Aguirre, and J. Quemada, "Facilitating the creation of K-12 interactive learning objects using a multi device web tool," in *Proceedings of the 2012 Frontiers in Education Conference (FIE 2012)*, 2012.